	Type	L #	Hits	Search Text	DBs	Time Stamp
1	BRS	L1	1	6532527.pn.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/03/16 10:54
2	BRS	L2	8	virtual$7 with conver$4 with request$3 with volume	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/03/16 11:07
3	BRS	L3	45	virtual$7 same conver$4 same request$3 same volume	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/03/16 11:08

US-PAT-NO: RE36989

DOCUMENT-IDENTIFIER: US RE36989 E
See image for Certificate of Correction

TITLE: Virtual storage system and method

DATE-ISSUED: December 12, 2000

US-CL-CURRENT: 711/118, 710/68 , 711/111 , 711/113 , 711/137 , 711/162
, 711/168

APPL-NO: 08/ 934732

DATE FILED: September 22, 1997

REISSUE-DATA:

US-PAT-NO	DATE-ISSUED	APPL-NO	DATE-FILED
04467421	August 21, 1984	384381	June 2, 1982

PARENT-CASE:

This application is a continuation-in-part of Ser. No. .[.261,950.]. .Iadd.06/261,951, now abandoned, .Iaddend.filed May 8, 1981, and of Ser. No. .[.085,909.]. .Iadd.06/085,909, now abandoned, .Iaddend.filed Oct. 18, 1979, both in the name of Barry B. White.

---------- KWIC ---------

Detailed Description Text - DETX (25):

FIG. 5 shows the overall layout of a data processing and storage system
utilizing the presently-preferred embodiment of the virtual storage system of
the invention. The particular system shown in FIG. 3 shows a pair of host
computers 60 and 61 each connected to a pair of virtual control processors
(VCPs) 74A and 74B which amount to the heart of the virtual storage system of
the invention. In the preferred embodiment the VCPs comprise Magnuson M80
computers, the main memories of which include both the cache and the address
memory space in which is stored the "directory" which lists the locations on
disk at which the various subportions of a user-defined sequential file are
stored. The VCPs 74A and 74B are each in turn connected to a data base, which
in the configuration shown each comprise a pair of disk controller units 64 and
65 each operating a pair of disk drives each 66 and 67, and 68 and 69,
respectively, and to a pair of tape controllers 70 and 71 having attached
thereto tape drives 72 and 73, respectively. The data bases may comprise
additional disk and tape units, depending on the capacity of the VCPs 74A and
74B. The virtual control processors 74A or 74B may each also have secondary

connections to the other's data base, for backup purposes. Accordingly when a host requests a specific user data set or "virtual volume" the request need only specify which of the two virtual control processors 74A or 74B controls the data base within which is stored that virtual volume. The virtual control processors 74A and 74B are each able, using their internal address store, to convert the name of the file into the location of the data on the disk or tape unit(s) involved and forward it to the host without further instruction from the host.

US-PAT-NO: 5761411

DOCUMENT-IDENTIFIER: US 5761411 A

TITLE: Method for performing disk fault prediction operations

DATE-ISSUED: June 2, 1998

US-CL-CURRENT: 714/47, 714/42 , 714/48

APPL-NO: 08/ 518831

DATE FILED: August 24, 1995

PARENT-CASE:

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a continuation-in-part of U.S. patent application Ser. No. 08/404,812 filed Mar. 13, 1995, entitled "Drive Failure Prediction Techniques for ATA Disk Drives", pending, assigned to the Assignee of the present application and hereby incorporated by reference as if reproduced in its entirety.

This application is also related to U.S. patent application Ser. No. 08/519,104 entitled "IDE Disk Fault Prediction Virtual Driver" and U.S. Ser.

No. 60/002,702 entitled "Hardware Component Interface for Desktop Management System", both of which were filed on even date herewith, assigned to the Assignee of the present application and hereby incorporated by reference as if reproduced in their entirety.

---------- KWIC ---------

Detailed Description Text - DETX (41):

The uppermost level of the layered block device driver 138 contains the file system driver 140. The file system driver 140 manages high-level I/O <u>requests</u> from the applications 132-1 through 132-N. Beneath the file system driver 140 are one or more upper level driver(s) 142, the exact number of which will vary based upon the configuration of the layered block device drivers 138. Typically, the upper level driver(s) 142 will carry out functions which include transitions of I/O <u>requests from a volume</u> orientation to a logical device orientation, from a logical device to a physical device orientation and from a physical device orientation to an adapter orientation. Drivers at higher levels generally deal with logical I/O operations while drivers at lower levels carry out physical I/O to adapters. Beneath the upper level driver(s) 142 in the call-down stack is the IDE DFP <u>virtual</u> driver 144. As will be

more fully described later, the IDE DFP virtual driver directs accesses from the various Windows 95 applications 132-1 through 132-N via the file system driver 140 and accesses from the DFP application 136 via the DEV IOCTL interface 148 directly to the IDE drive 134 while replies from the IDE drive 134 are selectively directed to either the DEV IOCTL interface 148 (if their destination is the DFP application 136) or the upper level driver(s) 140 if their destination is elsewhere, for example, a selected one of the applications 132-1 through 132-N.
The IDE DFP virtual driver 144 also monitors every command sent to the IDE port driver 146 from the file system driver 140 and records its completion. Thus, when an IDE command is sent to the IDE DFP virtual driver 144 from the file system driver 140, the command is passed to the IDE port driver 146 and a count of the total number of pending commands is incremented. Conversely, when a reply to the IDE command sent from the file system driver 140 is returned by the IDE port driver 146, the count of the total number of pending commands is decremented. When a DFP command is received from the DFP application 136, the IDE DFP virtual driver 144 will queue any later IDE commands from the file system driver 140 until a reply is received. If, however, an IDE command sent from the file system driver 140 is pending when the DFP command

is received,
the DFP command will be queued until replies to all of the pending IDE commands
are received.

US-PAT-NO: 5701486

DOCUMENT-IDENTIFIER: US 5701486 A

TITLE: Tracing technique for application programs using protect
mode addressing

DATE-ISSUED: December 23, 1997

US-CL-CURRENT: 717/128, 711/203 , 713/1

APPL-NO: 08/ 164665

DATE FILED: December 8, 1993

PARENT-CASE:

This a continuation of patent application Ser. No. 07/501,983 filed Mar.
29, 1990, now abandoned.

---------- KWIC ---------

Detailed Description Text - DETX (6):
Segment 220 checks to see if a device driver exists that is authorized to
convert a physical address to a virtual address and "pin" these addresses
together. If not, T1 remains equal to V1. If the device driver exists,

segment 220 calls the device driver by requesting a virtual address that is "pinned" to a physical address from device driver 50. In the preferred embodiment, device driver 50 has been told during the initialization of the computer system that physical address P was available to be pinned to a virtual address. Other embodiments may require segment 220 to pass a physical address P to device driver 50. In either event, Physical address P is contained in physical storage 20, and is usually a "safe place" to direct a write statement to, such as Read Only Memory (ROM) or Read Only Storage (ROS), where a destructive write will not occur but where a logic analyzer can detect the attempted write operation. Device driver 50 is given special authority by the operating system to convert physical address P to a virtual address V2, and performs this conversion. Device driver 50 tells the operating system that the application program "owns" this virtual address V2, and also prevents the operating system from changing the mapping between P and V2. These steps "pin" virtual address V2 to physical address P. In the preferred embodiment, device driver 50 uses a "PhysToUVirt" DevHelp function under OS/2 to pin virtual address V2 to physical address P. The "PhysToUVirt" DevHelp function is explained in more detail on pages 5-1, 5-6, 5-31, and 5-32 of "Operating

System/2 Programming Tools and Information Version 1.2: I/O Subsystems and
Device Support, Volume 1, Device Drivers", number 64F0282, First edition, Sep.
1989.

PAT-NO: JP403225417A

DOCUMENT-IDENTIFIER: JP 03225417 A

TITLE: INPUT/OUTPUT CONTROL SYSTEM FOR EXTENDED STORAGE DEVICE

PUBN-DATE: October 4, 1991

INVENTOR-INFORMATION:

NAME

TAKENAGA, SHINKICHI

YAGYU, KAZUHIKO

ASSIGNEE-INFORMATION:

NAME	COUNTRY
NEC CORP	N/A
TOHOKU NIPPON DENKI SOFTWARE KK	N/A

APPL-NO: JP02021344

APPL-DATE: January 30, 1990

INT-CL (IPC): G06F003/08, G06F012/00

ABSTRACT:

PURPOSE: To extend various system elements and to improve the utilization

efficiency of a whole system by performing batch control over request source identifiers and enabling dynamic assignment and releasing.

CONSTITUTION: An application program 6 actuates a virtual volume control means 5 by requiring the input/output against a virtual volume 2. The virtual volume control means 5 converts a virtual volume number whose input and output are required and an address in the virtual volume into an extended storage device number and an address in the extended storage device according to the information on the correspondence between the held virtual volume 2 and extended storage device 1. Then a request source identifier control means 4 is required for input and output with the converted extended storage device number and the address in the extended storage device, the request source identifier control means 4 performs the bath control over the request source identifier, which is assigned and released at the input/output requirement from a lower-layer volume control means 5. Consequently, the various system elements are extended and input/output operation is performed efficiently.

US-PAT-NO: 6615327

DOCUMENT-IDENTIFIER: US 6615327 B1

TITLE: Method and system for backing up data of data processing
devices including fixed length block format data
conversion to variable length block format

DATE-ISSUED: September 2, 2003

US-CL-CURRENT: 711/162, 360/48 , 711/112 , 714/7

APPL-NO: 09/ 506271

DATE FILED: February 17, 2000

PARENT-CASE:

CROSS-REFERENCE TO RELATED APPLICATION

This application is related to U.S. patent application Ser. No. 08/912,872
filed Aug. 19, 1997, now U.S. Pat. No. 6,115,797, the subject matter of
which is incorporated herein by reference.

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
JP	11-041048	February 19, 1999

---------- KWIC ---------

Detailed Description Text - DETX (21):
The backup software 112 generates volume information. This volume information is used to request the storage control processor to read the backup data and begin performing the backup operation (step 1104). The storage control processor 14 uses volume serial number identifying means 142 to determine if the specified volume serial number is within a numerical range associated with data from the external storage apparatus 5. If the specified volume serial number is not within a range associated with data from the external storage apparatus 5, the open-system volume in the external storage apparatus 4 is backed up. Thus, in this case operations similar to the one at step 1303 are performed (step 1105). If the specified volume serial number is within the range associated with data from the external storage apparatus 5, the storage control processor 14 uses the communication line 34 to request the open system 2 to read data. In response to this request, the open system 2 starts a backup program and reads backup data from the open-system volume. This data is transferred via the communication line 34 to the storage control processor 14 (step 1106). The storage control processor 14 uses the data format converter 141 to convert the data transferred from the

open system 2
into the variable-length block format while generating virtual C fields (step
1107). This allows the data to be used by the operating system 113 of the
mainframe 1. The storage control processor 14 sends the converted data to the
operating system 113. The data sent to the operating system 113 is stored in
the backup apparatus 3 by the backup software 112 (step 1108). The storage
control processor 14 then checks to see if there is any remaining backup data
from the open system 2. If there is backup data remaining, the operations from
step 1107 through step 1108 are repeated (step 1109). Once all the backup data
has been processed, the storage control processor 14 reports that the backup
operation has been completed. The open system 2 mounts the open-system volume
53 and resumes operations (step 1110).

US-PAT-NO: 6467054

DOCUMENT-IDENTIFIER: US 6467054 B1

TITLE: Self test for storage device

DATE-ISSUED: October 15, 2002

US-CL-CURRENT: 714/42, 369/53.17

APPL-NO: 09/ 258858

DATE FILED: February 26, 1999

PARENT-CASE:

CROSS REFERENCE TO RELATED APPLICATIONS

**This application is a continuation in part of U.S. patent application Ser.
No. 09/076,300 filed May 11, 1998, which is a continuation of Ser. No.
08/518,831, now U.S. Pat. No. 5,761,411 filed Aug. 24, 1995, which is a
continuation-in-part of U.S. Pat. No. 08/404,812 filed Mar. 13, 1995, now
abandoned, all assigned to the Assignee of the present application and hereby
incorporated by reference as if reproduced in its entirety.**

**This application is also related to U.S. patent application Ser. No.
09/259,393 entitled "Background Read Scanning with Reallocation"**

**and U.S.
patent application Ser. No. 09/259,622, entitled "Error Logging", both of
which were filed concurrently herewith, and U.S. Pat. No. 5,761,411, all
assigned to the Assignee of the present application and hereby incorporated by
reference as if reproduced in their entirety.**

**This application is also related to U.S. Pat. No. 5,828,583, which is a
continuation of U.S. patent application Ser. No. 08/404,812 filed Mar. 13,
1995 entitled "Drive Failure Prediction Techniques for ATA Disk Drives", now
abandoned, all of which are hereby incorporated by reference herein.**

---------- KWIC ---------

Detailed Description Text - DETX (72):

**The uppermost level of the layered block device driver 138 contains the file
system driver 140. The file system driver 140 manages high-level I/O <u>requests</u>
from the applications 132-1 through 132-N. Beneath the file system driver 140
are one or more upper level driver(s) 142, the exact number of which will vary
based upon the configuration of the layered block device drivers 138.
Typically, the upper level driver(s) 142 will carry out functions which include
transitions of I/O <u>requests from a volume</u> orientation to a logical**

device
orientation, from a logical device to a physical device orientation and from a
physical device orientation to an adapter orientation. Drivers at higher
levels generally deal with logical I/O operations while drivers at lower levels
carry out physical I/O to adapters. Beneath the upper level driver(s) 142 in
the call-down stack is the IDE DFP virtual driver 144. As will be more fully
described later, the IDE DFP virtual driver directs accesses from the various
Windows 95 applications 132-1 through 132-N via the file system driver 140 and
accesses from the DFP application 136 via the DEV IOCTL interface 148 directly
to the IDE drive 134 while replies from the IDE drive 134 are selectively
directed to either the DEV IOCTL interface 148 (if their destination is the DFP
application 136) or the upper level driver(s) 140 if their destination is
elsewhere, for example, a selected one of the applications 132-1 through 132-N.
The IDE DFP virtual driver 144 also monitors every command sent to the IDE port
driver 146 from the file system driver 140 and records its completion. Thus,
when an IDE command is sent to the IDE DFP virtual driver 144 from the file
system driver 140, the command is passed to the IDE port driver 146 and a count
of the total number of pending commands is incremented. Conversely, when a
reply to the IDE command sent from the file system driver 140 is

returned by
the IDE port driver 146, the count of the total number of pending commands is
decremented. When a DFP command is received from the DFP application 136, the
IDE DFP virtual driver 144 will queue any later IDE commands from the file
system driver 140 until a reply is received. If, however, an IDE command sent
from the file system driver 140 is pending when the DFP command is received,
the DFP command will be queued until replies to all of the pending IDE commands
are received.

US-PAT-NO: 6526478

DOCUMENT-IDENTIFIER: US 6526478 B1

TITLE: Raid LUN creation using proportional disk mapping

DATE-ISSUED: February 25, 2003

US-CL-CURRENT: 711/114, 711/171 , 711/209

APPL-NO: 09/ 496031

DATE FILED: February 2, 2000

---------- KWIC ---------

Detailed Description Text - DETX (44):

FIG. 6 illustrates the process of converting a logical block address to a segment number wherein segment number is a part of a proportionally mapped LUN. The proportionally mapped LUN was created substantially according to the algorithm described above with respect to FIG. 5. The process begins at 602 with the receipt of the logical block address from the host. Essentially, since the host views a virtual volume, such as virtual volume 50 shown in FIG. 1, the host uses logical block addresses to address specific files

and sends an
access <u>request</u> for a file or data and includes a logical block
address to
locate that portion of the file.